About AdPushup JavaScript

AdPushup's ad serving technology is delivered via a JavaScript code during page load. We call this **AdPushup JavaScript (APJS)**. Publishers often ask us about the speed and SEO impact of running APJS. We use a number of **code optimization techniques** to speed up the JS during generation, delivery, and execution. This includes the following:

1

Modular Code Generation

We don't bundle all the JavaScript together from the start. Instead, we have a **core JS module** which includes our layout optimization engine. Then, we include additional modules such as header bidding, new ad formats, and automatic in-content ad placement, based on what the publisher wants. This helps us **reduce the bundle size** and ensure that publishers only load the scripts they are actually using. In addition, **tree-shaking** is used for dead code elimination before run time, saving disk and CPU usage on the client-side.

2

CDN-based Delivery

Next, we use a CDN-based delivery for our JavaScript code. This means that instead of serving the JavaScript from our servers, we use a **high-speed content delivery network** which keeps mirror copies of our code. This CDN delivers the JavaScript to the publisher's website from a server nearest to their actual physical location. So for instance, if a website hosted in New York, the JS will also be delivered from a server in New York. This reduces delivery latency. Our CDN also uses **GZip compression** to further reduce bundle size.

3

Lazy Loading Script

Finally, we use lazy loading during code execution to ensure that only **essential JS assets are delivered** to the user's browser. Lazy loading is a technique that loads assets (scripts, files, images...) only when they are needed by the browser. This is opposed to synchronous or asynchronous delivery, where assets load regardless of whether or not they are needed on-page. We deploy lazy loading for (1) AdPushup JavaScript and (2) ad units served on publisher webpages, **improving load times** during code execution and ad delivery.

Page Speed & SEO Impact Analysis

In order to demonstrate the real-life impact of using APJS on a website's page speed and search performance, **we conducted a test on five websites within our publisher network**, our methodology is shared below. We encourage you to recreate the tests and verify the results.

Our Methodology

- We selected five publisher websites in our network at random
- Then we used Lighthouse, a page performance audit tool by developed by Google, to analyze the impact of APJS on page load speed and on-page SEO. Lighthouse can be used in multiple ways; we used it from the Chrome DevTools
- In the control group we have websites which are currently using AdPushup; in the experimental group, we have the same websites with AdPushup code disabled.

Re-creating the Test

- For the control group, open one of the websites that you wish to audit
- Open Chrome DevTools and navigate to the Audits tabSelect 'Desktop' for Device, check only 'Performance' for Audits and disable throttling; Run the test
- The results obtained are for the site's performance while using AdPushup; a detailed description of the metrics is given below
- To run the analysis for the experimental group, navigate to the network tab and filter the requests by the name of adpushup.js
- Right click on any such request and block the request URL
- Go back to the Audits tab and repeat the analysis under the same conditions
- Compare the results!

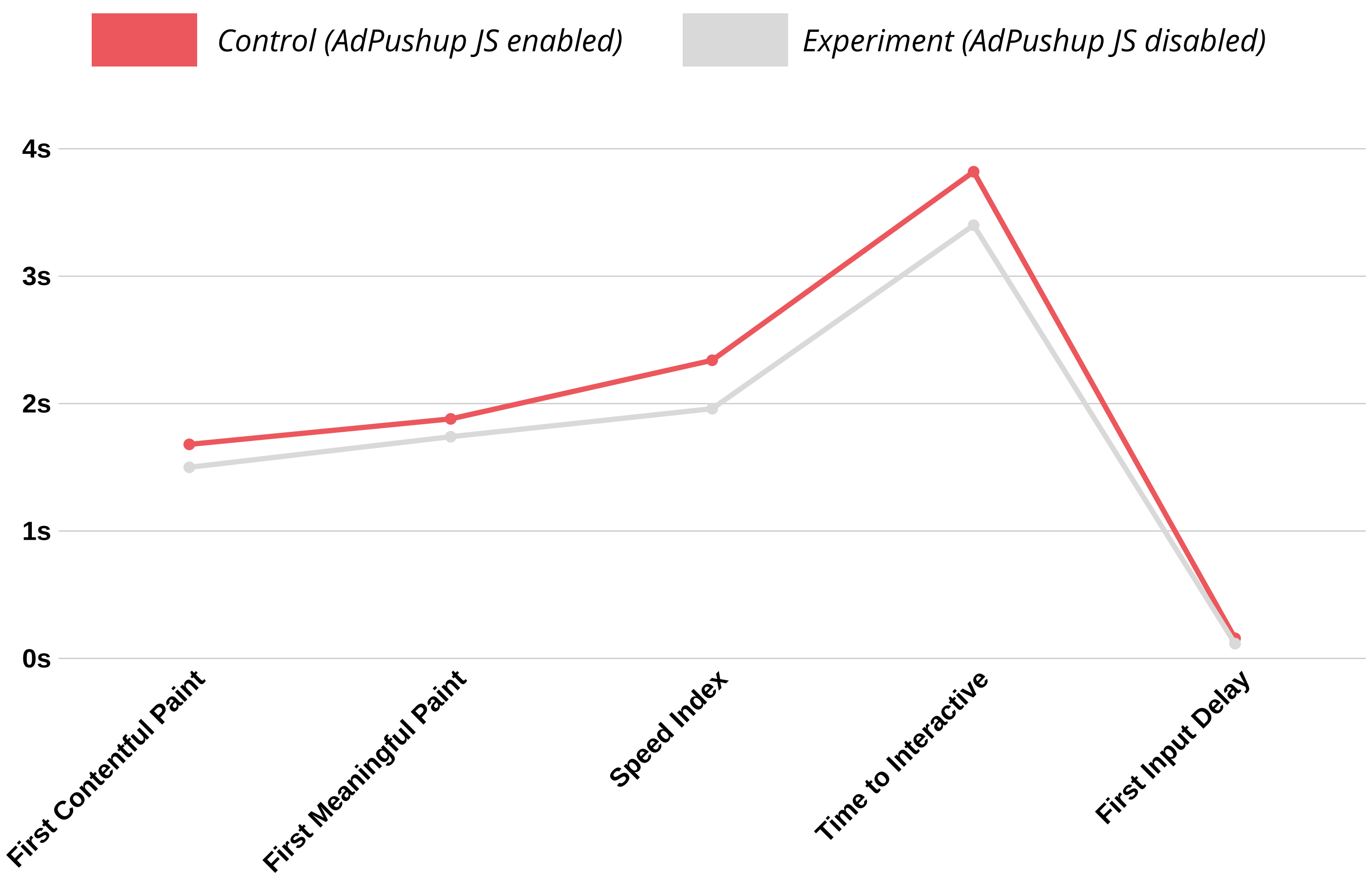
Metrics Used

- **First Contentful Paint:** First Contentful Paint marks the time at which the first text or image is painted.
- **Speed Index:** Speed Index shows how quickly the contents of a page are visibly populated.
- **Time to Interactive:** Time to interactive is the amount of time it takes for the page to become fully interactive.
- **First Meaningful Paint:** First Meaningful Paint measures when the primary content of a page is visible.
- **First CPU Idle:** First CPU Idle marks the first time at which the page's main thread is quiet enough to handle input.
- **Max Potential First Input Delay:** The maximum potential First Input Delay that your users could experience is the duration, in milliseconds, of the longest task.

Page Speed Results

Although Lighthouse yields a comprehensive page speed report consisting of six different metrics, the key ones are First Meaningful Paint and Time to Interactive.

- **First Meaningful Paint** is the time it takes for the primary content of a page to become visible. In our test, this value was higher by just 140ms on average (with the difference being 0s in three instances) for websites in the control group.
- **Time to Interactive**, measures the amount of time it takes for a page to become fully interactive, this value was found to be higher by ~400ms on average for websites in the control group, with one page with no change in TTI at all and two others with less than 200ms variation. Considering that the average TTI value was 3400ms in the experimental group, a 400ms change is a blink of an eye.
- **First Input Delay** is the time from when a user first interacts with a site to the time when the browser responds to that interaction. The change in FID upon adding APJS was a mere 40ms—a humanly imperceptible time difference.



Average values for key page speed parameters

On-page SEO Results

The test for SEO was carried out using the Lighthouse toolkit that measures the page's SEO performance based on certain predefined programmatic tests, including:

- Structured data is valid
- Has a <meta name="viewport"> tag with width or initial-scale
- Document has a <title> element
- Page has successful HTTP status code
- Links have descriptive text
- Page isn't blocked from indexingrobots.txt is valid
- Document has a valid hreflang
- Document has a valid rel=canonical
- Document avoids plugins
- Document uses legible font sizes
- Tap targets are sized appropriately

0% SEO Impact

As it can be seen, **these parameters are objective in nature**. Given that they have been cherry picked by Google to gauge the impact of on-page code on search performance, this is an ideal way to gauge the SEO impact of AdPushup's script.

Depending on whether a websites passes these tests or not, a cumulative weighted score is awarded to them. It's noteworthy that the **AdPushup JS script had zero impact** on each and every website's on-page SEO score in the study.

About AdPushup

AdPushup was formed in 2014 with a simple idea: While A/B testing was becoming popular, no one was using it to optimize publisher-side ad layouts. Our founders built and launched a prototype to get proof-of-concept, which resulted in double-digit revenue growth for our first website. Since then, we've garnered top media mentions, raised multiple rounds of funding, and expanded to being a one-stop revenue optimization platform for web publishers. In addition to being a Google NPM AdX Partner, we're a Microsoft Ventures backed startup, and winner of the NASSCOM Emerge 50 award. Today, we serve and optimize over 4 billion monthly ad impressions for our 300+ publishing partners.

Microsoft Partner | 2019 Partner of the Year Finalist Media & Communications Award

Winner Ad Network of the Year AgencyCon 2019

iab. member

Emerge 50 Awards

tag REGISTERED